REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
02-29-2012

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
7/1/09-11/30/11

4. TITLE AND SUBTITLE ICPL: Intelligent Cooperative Planning and Learning for Multi-agent Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-09-1-0522

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jonathan P. How, E. Frazzoli, and N. Roy

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of
Technology
77 Massachusetts Ave
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0686

12. DISTRIBUTION / AVAILABILITY STATEMENT

A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research objective was to develop a new planning approach for teams of multiple UAVs that tightly integrates learning and cooperative control algorithms at multiple levels of the planning architecture. The research results enabled a team of mobile agents to learn to adapt and react to uncertainty in situational awareness and unforeseen future events and thus successfully complete their missions in geographically extended and uncertain theaters of operation. Among application areas, we considered learning approaches for persistent patrolling games, in which one class of agents places point targets in a given region, and a second class of agents seeks to minimize the time necessary to discover such targets, using limited-range sensors. We analyzed equilibrium strategies in this class of problems and their stability. Our efforts provide a fundamental theory and architecture for the design of intelligent cooperative control systems for heterogeneous teams and the demonstration of the value of the theory through software and hardware experiments.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT

b. ABSTRACT

c. THIS PAGE

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES
20

19a. NAME OF RESPONSIBLE PERSON
Acia Adams-Heath

19b. TELEPHONE NUMBER *(include area code)*
(617) 715-4294

ICPL: Intelligent Cooperative Planning and Learning for Multi-agent Systems

AFOSR # FA9550-09-1-0522

Jonathan P. How, Emilio Frazzoli, and Nicholas Roy
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Abstract

The research objective was to develop a new planning approach for teams of multiple UAVs that tightly integrates learning and cooperative control algorithms at multiple levels of the planning architecture. The research results enabled a team of mobile agents to learn to adapt and react to uncertainty in situational awareness and unforeseen future events and thus successfully complete their missions in geographically extended and uncertain theaters of operation. Among application areas, we considered learning approaches for persistent patrolling games, in which one class of agents places point targets in a given region, and a second class of agents seeks to minimize the time necessary to discover such targets, using limited-range sensors. We analyzed equilibrium strategies in this class of problems and their stability. Our efforts provide a fundamental theory and architecture for the design of intelligent cooperative control systems for heterogeneous teams and the demonstration of the value of the theory through software and hardware experiments.

1 Research Summary

The following list provides the contributions of our research.

1. **Incremental Feature Dependency Discovery (iFDD)**: We introduced iFDD as a general approach for expanding a linear function approximation of a value function from an initial set of binary features to a more expressive representation that incorporates feature conjunctions. Our algorithm is simple to implement, fast to execute, and can be combined with any online reinforcement learning technique that provides an error signal. We provided asymptotic and rate of convergence analysis for iFDD when combined with TD learning. Furthermore, we empirically showed that iFDD can scale to UAV mission planning problems with hundreds of millions of state-action pairs where other adaptive methods do not scale [1, 2].

2. **Intelligent Cooperative Control Architecture (iCCA)**: We introduced iCCA as a framework for learning and adapting cooperative control strategies in real-time stochastic domains. The framework allows cooperative planners to guide learners to find good policies with less number of samples, while mitigating the risk involved in pure learning strategies. We extended our framework in three areas: 1) integrated learning methods with implicit policy forms, 2) supported stochastic risk models to realize probabilistic safety, and 3) enabled adaptive modeling of the system dynamics. We successfully demonstrated the performance of our approach by simulating limited-fuel UAVs aiming for stochastic targets in mission planning scenarios involving uncertainty [3–7].

3. **Data-Limited Model-Based Reinforcement Learning** We introduced a novel method for batch reinforcement learning with limited amount of data. Rather than finding the model through the maximum likelihood concept, our approach directly searches in the space of possible models that result in capable policies. We demonstrated the applicability of the new approach in a physical hydrodynamic card-pole problem with a very complex physics model.

4. **Incremental Sampling-based Algorithms for Planning and Learning Under Uncertainty** We studied a zero-sum game formulation of a dynamic vehicle routing problem: a system planner seeks to design dynamic routing policies for a team of vehicles to minimize the average waiting time of demands that are strategically placed in a region by an adversarial agent with unitary capacity operating from a depot. We characterized an equilibrium in the limiting case where vehicles travel arbitrarily slower than the agent (heavy load). We showed that such an equilibrium is constituted by a routing policy based on performing successive TSP tours through outstanding demands and a unique power-law spatial density centered at the depot location [8].

5. **Learning in Persistent Search: A Constrained Game Approach** We considered a class of dynamic vehicle routing problems, known as persistent search [9], in which a vehicle with limited sensor range aims to detect targets that arrive dynamically over time. A common assumption made in such settings [10–12] is that the distribution of arrivals is known a priori to the searcher. Both the analysis and tools that have been developed previously have heavily relied on the knowledge of the distribution. In our recent work [8], we relax some of these assumptions and provide performance guarantees by modeling the problem in the language of constrained games. To do so, we assume that the searcher faces an environment in which the samples are distributed in an adversarial manner that is consistent with history. In other words, we find the worst-case model of nature that can explain the past observations and find a search policy that is optimal against such adversarial model. The resulting optimization objective is non-convex, but can be convexified in special cases and, in general, has a special structure that allows good approximations with SOS methods [13]. The resulting solution can, under mild assumptions, guarantee a minimum level of performance.

2 Online Discovery of Feature Dependencies

Optimal planning under uncertainty is a challenging problem facing practitioners dealing with real-world domain. MDPs [14] facilitate a mathematical framework for solving these problem, but unfortunately for realistic multi-agent planning, the size of the state space is exponential in the number of agents and researchers quickly realized [15] the limitation of tabular representations and introduced approximations to cope with the complexity and boost the learning speed through generalization. Finding the “right” representation is a critical milestone to scale the existing MDP solvers to larger domains. Due to their ease of use, theoretical analytical, and empirical results, the linear family of function approximators have been favored in the literature [15, 16]. In this setting, the target function is approximated as the linear combination of a set of feature vectors. Many approaches try to find the right set of features offline [17, 18], but online methods enjoy the advantage of adaptability to dynamic environment and often the case lower computational complexity [19]. Hence these methods have improved the learning speed of existing reinforcement learning (RL) algorithms in low dimensional domains, yet existing online expansion methods do not scale well to high dimensional problems. Our research has explored the conjecture that one of the main difficulties limiting this scaling is that features defined over the full-dimensional state space often generalize poorly. Hence, we introduced *incremental Feature Dependency Discovery* (iFDD) as a computationally-inexpensive method for representational expansion that can be combined with any online, value-based RL method that uses binary features. Unlike other online expansion techniques, iFDD creates new features in low dimensional subspaces of the full state space where the approximation error persist.

The iFDD algorithm gradually captures nonlinearities within the linear approximation framework by introducing feature conjunctions as new binary features. We showed that, especially in high-dimensional domains, gradually adding feature dependencies (*e.g.* features corresponding to low dimensional subspaces of the full state space), encourages early generalization, which can speed up learning. The algorithm begins by building a linear approximation to the value function online using the initial set of binary features. It tracks the sum of absolute value of the approximation errors for all simultaneously activated feature pairs. We term the conjunction of each tracked feature pair as a *potential* feature and the cumulative approximation error associated with it as *relevance*. Once a potential feature's relevance exceeds a user-defined threshold, iFDD *discovers* that feature as a new binary feature, thus capturing the nonlinearity between the corresponding feature pair. The algorithm proceeds in three steps:

1. Identify *potential* features that can reduce the approximation error,

2. Track the relevance of each potential feature, and

3. Add potential features with relevance above a discovery threshold to the pool of features used for approximation.

Fig. 1 shows iFDD in progress. The circles represent initial features, while rectangles depict conjunctive features. The relevance of each potential feature f, ψ_f, is the filled part of the rectangle. The discovery threshold ξ, shown as the length of rectangles, is the only parameter of iFDD and controls the rate of expansion. This parameter is domain-dependent and requires expert knowledge to set appropriately. However, intuitively lower values encourage faster expansion and improve the convergence to the best possible representation, while higher values slow down the expansion and allow for a better exploitation of generalization. While the ideal value for ξ will depend on the stochasticity of the environment, we found our empirical results to be fairly robust to the value of the discovery threshold.

We focus on iFDD integrated with TD learning, but any on-line, value-based RL method could supply the feedback error. Notice that, if the initial features are such that no function approximation – linear or nonlinear – can satisfactorily approximate the underlying value function, then applying iFDD will not help. For example, if a key feature such as an agent's location is not included in the initial set of features, then the value function approximation will be poor even after applying iFDD.

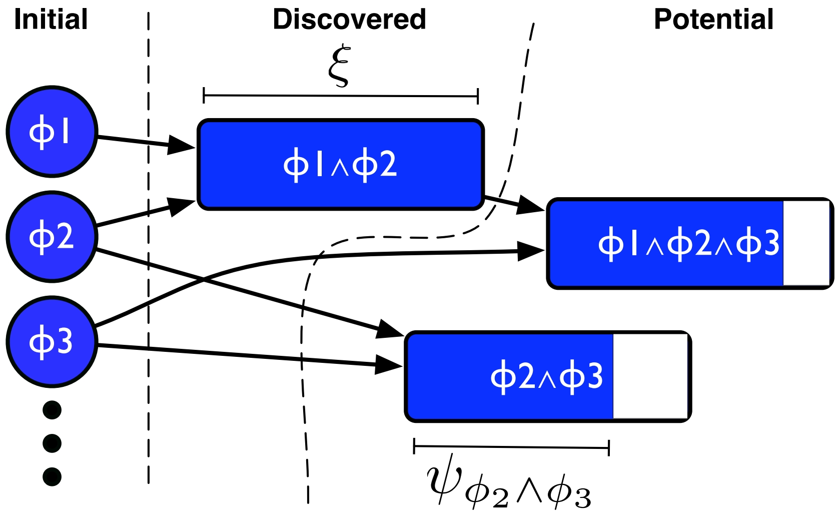


Fig. 1: A snapshot of iFDD: Initial features are circles, conjunctive features are rectangles. The *relevance* ψ_f of a potential feature f is the filled part of the rectangle. Potential features are discovered if their relevance ψ reaches the discovery threshold ξ.

2.1 Algorithm Details

The process begins with an initial set of binary features; let $\mathbf{F}$ be the current set of features used for the linear function approximation at any point in time. We use $\phi_f(s) = 1$ to indicate that feature $f \in \mathbf{F}$ is *active* in state s. After every interaction, we compute the local value function approximation error δ_t (*e.g.* the TD error), the current feature vector $\phi(s_t)$, and update the weight vector θ (in the TD case, $\theta_{t+1} = \theta_t + \alpha_t \delta_t \phi(s_t)$, where α_t is the learning rate). Next, Algorithm 1 is applied to discover new features.

The first step in the discovery process (lines 1,2) identifies all conjunctions of active features as potential features.[1] Considering only conjunctive features is sufficient for iFDD to converge to the best approximation possible given the initial feature set; conjunctive features also remain sparse and thus keep the per-time-step computation low. The relevance ψ_f of each potential feature $f = g \wedge h$ is then incremented by the absolute approximation error $|\delta_t|$ (line 4). If the relevance ψ_f of a feature f exceeds the discovery threshold ξ, then feature f is added to the set $\mathbf{F}$ and used for future approximation (lines 5,6).

The computational complexity of iFDD can be reduced through a sparse summary of all active features. Note that if feature $f = g \wedge h$ is active, then features g and h must also be active. Thus, we can greedily consider the features composed of the largest conjunction sets until all active initial features have been included to create a sparse set of features that provides a summary of all active features.[2] For example, if initial features g and h

[1] Conjunctions are stored in a "flat" representation, so there is only one conjunctive feature $a \wedge b \wedge c$ for the conjunction of features $a \wedge (b \wedge c)$ and $(a \wedge b) \wedge c$.

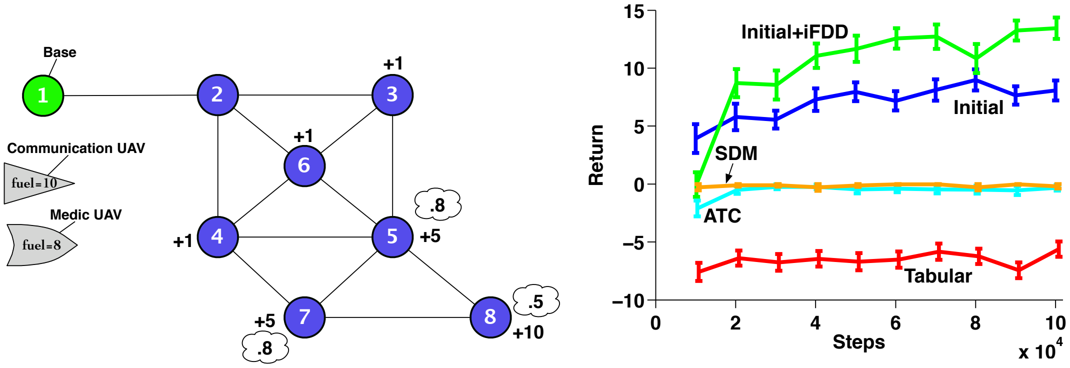
[2] Finding the minimum covering set is NP-complete but greedy selection gives the best polynomial time approximation.

```
Algorithm 1: Discover
Input: $\phi(s), \delta_t, \xi, \mathbf{F}, \psi$
Output: $\mathbf{F}, \psi$
foreach $(g, h) \in \{(i, j)|\phi_i(s)\phi_j(s) = 1\}$ do
    $f \leftarrow g \wedge h$
    if $f \notin \mathbf{F}$ then
        $\psi_f \leftarrow \psi_f + |\delta_t|$
        if $\psi_f > \xi$ then
            $\mathbf{F} \leftarrow \mathbf{F} \cup f$
```

```
Algorithm 2: Generate Feature Vector ($\phi$)
  Input: $\phi^0(s), \mathbf{F}$
  Output: $\phi(s)$
1 $\phi(s) \leftarrow \bar{0}$
2 $\mathit{activeInitialFeatures} \leftarrow \{i|\phi^0_i(s) = 1\}$
3 $\mathit{Candidates} \leftarrow$ SortedPowerSet($\mathit{activeInitialFeatures}$)
4 while $\mathit{activeInitialFeatures} \neq \emptyset$ do
5     $f \leftarrow \mathit{Candidates}$.next()
6     if $f \in \mathbf{F}$ then
7         $\mathit{activeInitialFeatures} \leftarrow \mathit{activeInitialFeatures} \setminus f$
8         $\phi_f(s) \leftarrow 1$
9 return $\phi(s)$
```

are active in state s and feature $f = g \wedge h$ has been discovered, then we set the $\phi_f(s) = 1$ and $\phi_g(s), \phi_h(s) = 0$ since g and h are covered by f. Algorithm 2 describes the above process more formally: given the initial feature vector, $\phi^0(s)$, candidate features are found by identifying the active initial features and calculating its power set ($\wp$) sorted by set sizes (lines 2,3). The loop (line 4) keeps activating candidate features that exist in the feature set $\mathbf{F}$ until all active initial features are covered (lines 5-8). In the beginning, when no feature dependencies have been discovered, this function simply outputs the initial features.

Using the sparse summary also can help speed up the learning process. Suppose there are two features g and h that, when jointly active, result in high approximation errors. However, if one of them is active, then the approximation error is relatively low. Let f be the discovered feature $f = g \wedge h$. In our sparse summary, when g and h are both active in the initial representation, we set $\phi_f = 1$ and $\phi_g, \phi_h = 0$. If only one is active, then $\phi_f = 0$. Only non-zero features contribute to the value function approximation, so the learning update rule updates θ_f only if both $g = 1$ and $h = 1$. Otherwise, θ_f remains unchanged when θ_g or θ_h are updated. By separating the learning process for the states in which the feature conjunction $f = g \wedge h$ is true from states in which only one of the features g or h is true, we can improve our value function approximation estimates for the more specific cases without affecting the generalization in other states. The iFDD algorithm initializes the coefficient for a new feature f as $\theta_f = \theta_g + \theta_h$, so that the value function approximation remains unchanged



(a) Heterogeneous UAV Mission Scenario

(b) Empirical Results

Fig. 2: Heterogeneous UAV scenario and the step based performance of the Sarsa method with feature discovery (green), without feature discovery (blue), and using the full representation from the beginning (red).

when first adding a feature. We provided asymptotic and guaranteed rate of convergences analysis together with the computational complexity guarantees for iFDD [1, 2].

The usability of our approach in large state space such as UAV mission planning was confirmed by comparing the effectiveness of iFDD with Sarsa [14] against representations that (i) use only the initial features, (ii) use the full tabular representation, and (iii) use two state-of-the-art representation-expansion methods: adaptive tile coding (ATC), which cuts the space into finer regions through time [20], and sparse distributed memories (SDM), which creates overlapping sets of regions [21]. Fig. 2(a) depicts a rescue mission in which a heterogeneous team of UAVs plan to rescue as many people as possible highlighted as positive numbers close to each node. To carry out the rescue at a particular node, the medic UAV should visit the node while the communication UAV is no further than one edge from that node in order to provide satellite information. For nodes with stochastic rescue outcomes, the numbers inside clouds represent the success rate. Planning space for both of these domains exceeds hundred million possibilities. Fig. 2(b) shows the performance of all representation techniques, based on the number of interaction steps with the system. Adding useful features, iFDD allowed the agent to learn the task substantially better than the other methods. State-of-the-art method guided the agent to believe that in both domains sending out UAVs is dangerous and should be avoided, hence the zero performance [1].

In the bigger picture, our theoretical results provided a fundamental insight on why moving from a coarse to fine representation is a sound approach for learning, which can explain empirical observations in both computer science and brain and cognitive science communities. In particular, our findings can explain the empirical results of Whiteson *et al.* [20], in which starting with one big feature and adaptively increasing the number of features, provided faster learning compared to starting with a fixed learned representation. At the same time, our theoretical results shed light on the observation of why human subjects first used coarse features for a task of classification and later used finer features as they gained more experience [22].

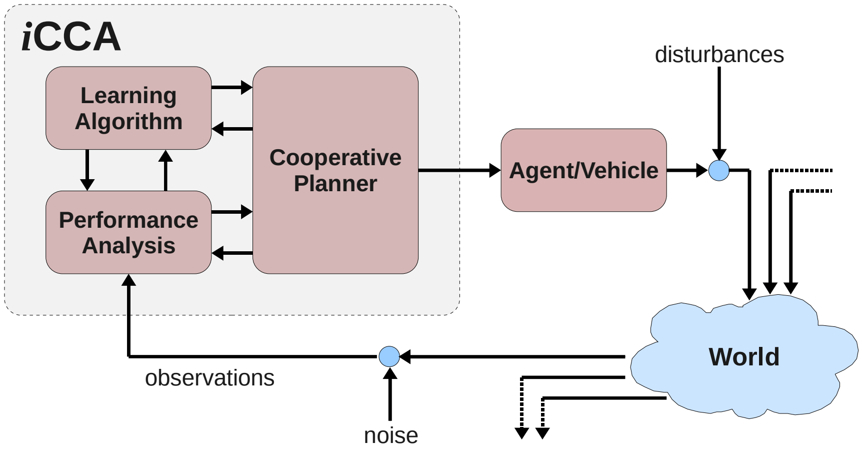


Fig. 3: **i**ntelligent **C**ooperative **C**ontrol **A**rchitecture

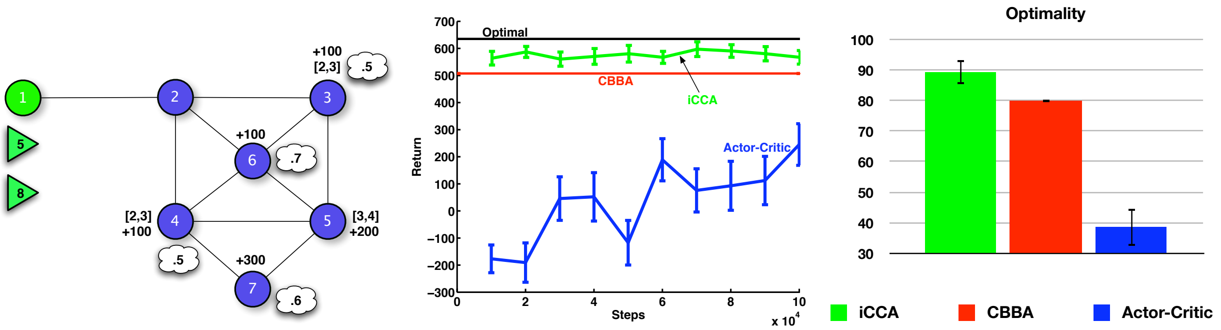
3 Merging Cooperative Planning and Learning

We developed a constructive relationship between a cooperative planner and a learner to mitigate the learning risk while boosting the asymptotic performance and safety of agent behavior. Our approach is an instance of the intelligent cooperative control architecture (iCCA) in Fig. 3 [4, 5]. The iCCA facilitates a synergistic integration of (i) cooperative planners that provide safe baseline capability for achieving challenging multi-agent mission objectives and (ii) learning algorithms that can improve the long-term performance of the system in real-world applications. In this framework, a learner initially follows a "safe" policy generated by a cooperative planner. The learner incrementally improves this baseline policy through interaction, while the performance analysis module avoids behaviors deemed to be "risky".

3.1 Stage 1

In the first stage of this research, we considered the most basic components needed to realize the iCCA framework. In particular, we assumed that **I)** the risk model is deterministic, **II)** the learning algorithm has an explicit parameterization for the policy which can be directly adjusted, and **III)** the approximated dynamics model is fixed [4]. As a result given the model of the system, a single simulated trajectory was sufficient to predict the risk involved in following a certain policy. In order to guide the learner early in the process, we initialized the learner's policy parameters in a way to match the planner's policy. Later on the learning algorithm could deviate from the planner's policy by reflecting back on the past experiences.

We verified the applicability of the iCCA framework for the UAV scenarios shown in Figure 4-(a), where a team of homogeneous UAVs carry out a mission which involves stochasticity. The green node represents the base, while blue nodes indicate target locations. Rewarding locations are tagged with a positive number. Notice that some nodes have time constraints shown as limits in square brackets. At the same time visiting a location by itself does not guarantee the completion of the task. Cloud figures on the top of each location dictates the probability of success for a UAV visiting the corresponding node. Finally the fuel capacity of each UAV is limited and the mission would fail with huge penalty if any of the



(a) UAV Mission Scenario

(b) Step based performance

(c) Optimality after training

Fig. 4: (a) Mission scenarios of interest: A team of two UAVs plan to maximize their cumulative reward along the mission by cooperating to visit targets. Target nodes are shown as circles with rewards noted as positive values and the probability of receiving the reward shown in the accompanying cloud. Note that some target nodes have no value. Constraints on the allowable visit time of a target are shown in square brackets, (b,c) the step based performance of the CBBA, Actor-Critic and CBBA.

UAVs fail to be at the base by the end of the mission horizon. In particular, we took advantage of the consensus-based bundle algorithm (CBBA) [23] for the planner, while using the Natural Actor-Critic method [24] for the learning module. 4-(b) depicts the performance of the optimal solution calculated using backward dynamic programming, CBBA, actor-critic and iCCA based on the number of interactions with the domain. Results are averaged over 30 runs. We can see the advantage of the iCCA method over its individual components: CBBA and actor-critic. This is due to the fact that iCCA explores “interesting” areas of the state-space while CBBA covers for the cases where actor-critic component can not provide a “safe” action. Finally, Figure 4-(c) shows the optimality of each method measured by the dynamic programming solution. Putting both learning and planning components together, iCCA framework could boost the performance of the best individual component by about 10%.

3.2 Stage 2

The second stage of this research [3] relaxed assumption **I**: the iCCA framework integrated learning schemes with implicit policy forms (*e.g.* SARSA [25]). The idea is motivated by the concept of the R_{max} algorithm [26]. The approach can be explained through the mentor-protégé analogy, where the planner takes the role of the mentor and the learner takes the role of the protégé. In the beginning, the protégé does not know much about the world, hence, for the most part s/he takes actions advised by the mentor. While learning from such actions, after a while, the protégé feels comfortable about taking a self-motivated actions as s/he has been through the same situation many times. Seeking permission from the mentor, the protégé could take the action if the mentor thinks the action is safe. Otherwise the protégé should follow the action suggested by the mentor.

Algorithm 3 details the new cooperative learning process. On every step, the learner inspects the suggested action by the planner and estimates the “knownness” of the state-

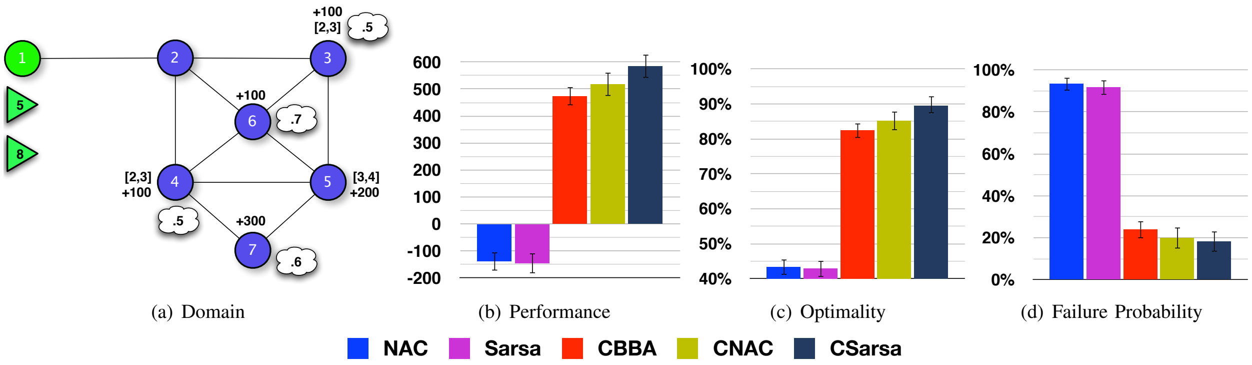
```
Algorithm 3: Cooperative Learning-2
   Input: $s, r$
   Output: $a$
 1 $a \sim \pi^p(s)$                                   /* CooperativePlanner */
 2 knownness $\leftarrow \min\{1, \frac{\texttt{count}(s,a)}{\mathcal{K}}\}$
 3 if $rand() <$ knownness then
 4     $a' \sim \pi^l(s)$                                /* Learner */
 5     if $safe(s, a')$ then
 6         $a \leftarrow a'$
 7 else
 8     count$(s, a) \leftarrow$ count$(s, a) + 1$
 9 learner.update$(s, r, a)$
10 return $a$
```

action pair by considering the number of times that state-action pair has been experienced following the planner's suggestion. The $\mathcal{K}$ parameter controls the transition speed from following the planner's policy to following the learner's policy. Given the knownness of the state-action pair, the learner probabilistically decides to select an action from its own policy. If the action is deemed to be safe, it is executed. Otherwise, the planner's policy overrides the learner's choice (lines 4-6). If the planner's action is selected, the knownness count of the corresponding state-action pair is incremented. Finally the learner is executed depending on the choice of the learning algorithm. Note that any control RL algorithm, even the Actor-Critic family of methods, can be integrated with cooperative planners using Algorithm 3 as line 9 is the only learner-dependent line, defined in the general form.

We empirically tested the new extension by combining CBBA [23] with Sarsa [25] and Natural Actor-Critic (NAC) [24]. Resulting methods are named CSarsa and CNAC respectively. Fig. 5-(a) shows the mission scenario of interest: a team of two limited fuel UAVs plan to maximize their cumulative reward along the mission by cooperating to visit targets. Target nodes are shown as circles with rewards noted as positive values and the probability of receiving the reward shown in the accompanying cloud. Note that some target nodes have no value. Constraints on the allowable visit time of a target are shown in square brackets. Fig. 5-(b,c) shows the result of NAC, Sarsa, CBBA, CNAC, and CSarsa algorithms at the end of the training session in the UAV mission planning scenario. Cooperative learners (CNAC, CSarsa) performed very well with respect to overall reward and risk levels, improving the performance of the baseline CBBA planner up to 30%. Finally, Fig. 5-(d) depicts the risk involved in executing each of the approaches after the learning phase. As expected pure learning strategies (NAC, Sarsa) has more than 90% risk as they do not restrict their exploration. While executing CBBA involves about 25% risk, cooperative learners could increase the reliability of the system up to 8%.

3.3 Stage 3

In the final stage, we closed the feedback loop in our system completely, by relaxing assumptions **II** and **III**, allowing the risk analysis module to support stochastic risk models and the



(a) Domain (b) Performance (c) Optimality (d) Failure Probability

Fig. 5: (a) Mission scenarios of interest, (b,c,d) Performance, optimality, and Risk of NAC, Sarsa, CBBA, CNAC, and CSarsa algorithms at the end of the training session

system dynamics model to be adaptive [6, 7]. To relax assumption **II**, given a fixed state, the risk associated with a proposed action by the learner is estimated using Monte-Carlo sampling. The user defines a certain threshold, where actions with expected risk more than this threshold are filtered. Assumption **III** is relaxed by allowing the model, incrementally adjusting its parameter as the agent interacts with the system. This means given new sampled trajectories, not only the learner changes its corresponding policy, but also the model dynamics can adapt to the observed data to become more accurate. Once the accumulated change to the model becomes significant, the cooperative planner replans accordingly.

To investigate the effectiveness of our new approach, we added 5% movement noise for each UAV in the mission shown in Figure 5-a. This means moving along each edge has 95% chance of success and 5% chance of staying in the same node. As for the baseline cooperative planner, CBBA [23] was implemented in two versions: aggressive and conservative. The aggressive version used all remaining fuel cells in one iteration to plan the best set of target assignments ignoring the possible noise in the movement. Algorithm 4 illustrates the conservative CBBA algorithm. The input to the algorithm is the collection of UAVs (U). First the current fuel of UAVs are saved and decremented by 3 (lines 1-2). Then on each iteration, CBBA is called with the reduced amount of fuel cells. Consequently, the plan will be more conservative compared to the case where all fuel cells are considered. If the resulting plan allows all UAVs to get back to the base safely, it will be returned as the solution. Otherwise, UAVs with no feasible plan ($i.e.$ $Plan[u] = \o$) will have their fuels incremented, as long as the fuel does not exceed the original fuel value (line 8). Notice that aggressive CBBA is equivalent to calling CBBA method on line 5 with the original fuel levels. The iCCA algorithm with static model only took advantage of the conservative CBBA because the noise assumed to be fixed at 40%. As for iCCA with adaptive model (AM-iCCA), the planner switched from the conservative to the aggressive CBBA, whenever the noise estimate dropped below 25%.

Figures 6 shows the results of learning methods (SARSA, iCCA, and AM-iCCA) together with two variations of CBBA (conservative and aggressive) applied to the UAV mission planning scenario. Figure 6(a) represents the solution quality of each learning method after 10^5 steps of interactions. The quality of fixed CBBA methods were obtained through averaging

Algorithm 4: Conservative CBBA

```
Input: UAVs
Output: Plan
MaxFuel ← U.fuel
UAVs.fuel ← UAVs.fuel − 3
ok ← False
while not $ok$ or MaxFuel = UAVs.fuel do
    Plan ← CBBA(UAVs)
    ok ← True
    for $u \in$ UAVs, Plan[$u$] = $\emptyset$ do
        UAVs.fuel[$u$] ← min(MaxFuel[$u$], UAVs.fuel[$u$] + 1)
        ok ← False
return Plan
```

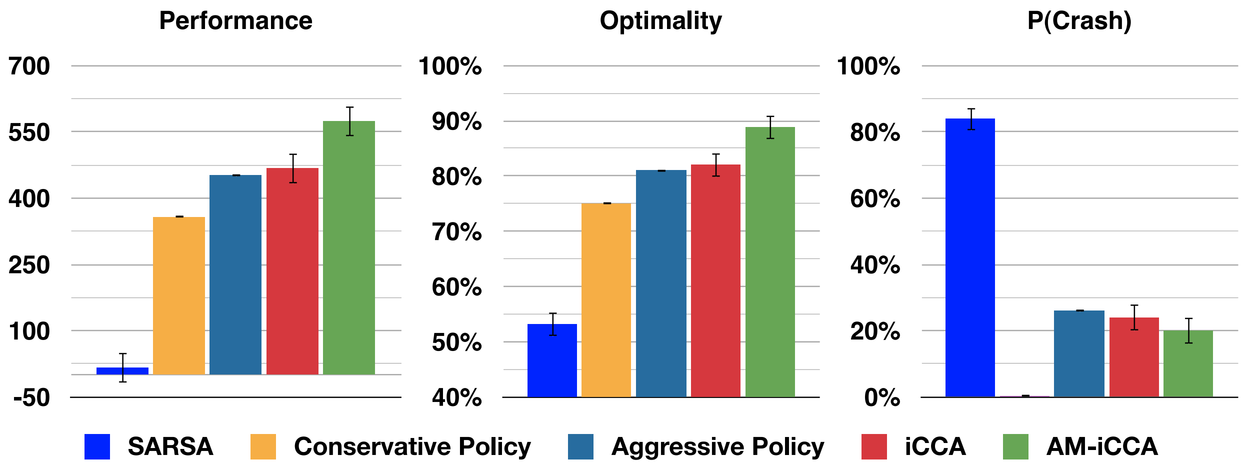


Fig. 6: Results of SARSA, CBBA-conservative, CBBA-Aggressive, iCCA and AM-iCCA algorithms at the end of the training session in the UAV mission planning scenario. AM-iCCA improved the best performance by 22% with respect to the allowed risk level of 20%.

over $10,000$ simulated trajectories, where on each step of the simulation a new plan was derived to cope with the stochasticity of the environment. Figure 6(b) depicts the optimality of each solution, while Figure 6(c) exhibits the risk of executing the corresponding policy. First note that SARSA at the end of training yielded 50% optimal performance, together with more than 80% chance of crashing a UAV. Both CBBA variations outperformed SARSA. The aggressive CBBA achieved more than 80% optimality in cost of 25% crash probability, while conservative CBBA had 5% less performance, as expected, it realized a safe policy with rare chances of crashing. The iCCA algorithm improved the performance of the conservative CBBA planner again by introducing risk of crash around 20%. While on average it performed better than that aggressive policy, the difference was not statistically significant. Finally AM-iCCA outperformed all other methods statistically significantly, obtaining close to 90% optimality. AM-iCCA boosted the best performance of all other methods by 22% on average (Figure 6-a). The risk involved in running AM-iCCA was also close to 20%, matching the selected risk threshold value.

4 Data-Limited Model-Based Reinforcement Learning

In model-based reinforcement learning (MBRL) an explicit dynamics model of the world is used to compute the policy. When the world dynamics model is unknown, as is often the case, we can attempt to infer the optimal policy by collecting data about the world through interactions with the environment. However, many real-world domains suffer from being *data-limited*, where the cost of collecting data through interaction limits how much training data is available to be used to build a dynamics model. When the true model is unknown in MBRL, we typically assume a model class. Unfortunately, for some real-world applications we cannot use the model class that contains the true world dynamics because this model class is either unknown or computationally intractable. This can be overcome by using an extremely large model ($\mathcal{M}$), allowing us to capture complex world dynamics. However, accurately fitting the parameters for these expressive model classes require a large amount of training data.

We investigate two methods of model estimation in data-limited MBRL. First, we use *reduced-order* dynamics models in order to effectively estimate the model parameters from limited data. A reduced-order model class, $\tilde{\mathcal{M}}$, is relatively small in comparison to the model class containing the true world dynamics. Fitting the reduced-order model parameters, θ, using a maximum likelihood approach can result in poor performance because the value of states under such model can be far from their real value. Instead of solving for the values for a given model $m \in \mathcal{M}$, we use the training data to estimate state values and then select the model that results in the highest true expected reward. This is done by combining a Monte-Carlo-like policy evaluation [27] for policy evaluation with a model gradient approach for policy improvement.

The quantity we are interested in estimating, $V(\pi^m; m_{true})$, is the expected future reward (value) of taking policy π^m in the world m_{true}. The typical approach for estimating $V(\pi^m; m_{true})$ is Monte Carlo (MC) simulation, where π^m is run in the true world to calculate the estimate $\hat{V}(\pi^m; m_{true})$. This method of estimating the value of a policy based on the accumulated reward from running it in the true world is called on-policy evaluation. On-policy evaluation of all the policies π^m for $m \in \hat{\mathcal{M}}$ is impossible since we generally have an infinitely large model class. Importance sampling (IS) is a method of MC simulation for off-policy evaluation. In IS, each trajectory's accumulated reward is weighted by the trajectory's likelihood under the policy we are evaluating. In the standard IS approach, any sampled trajectory from the data that took an action which did not agree with the action from π^m will not have an influence on the computation of $\hat{V}(\pi^m; m_{true})$. We initially assume that an arbitrary policy π^{train} was taken during training, resulting in π and π^{train} rarely agreeing on all actions taken throughout a trajectory and therefore a great deal of wasted data and a high variance estimate of $\hat{V}(\pi^m; m_{true})$.

Minimizing the variance of $\hat{V}(\pi^m; m_{true})$ is equivalent to requiring less data to accurately estimate $V(\pi^m; m_{true})$. To reduce the variance of $\hat{V}(\pi^m; m_{true})$ using IS we leverage two insights.

- We can prevent a trajectory from having zero probability by replacing any portion of

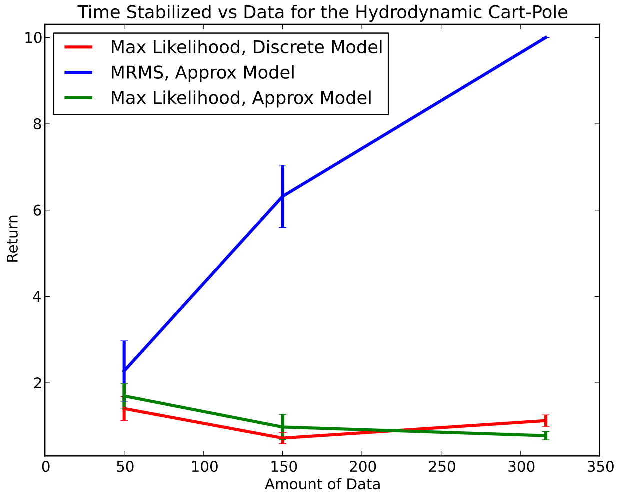


Fig. 7: Time stabilized versus number of episodes for the approximate model class fit using maximum likelihood and MRMS are shown in red and blue, respectively. The discrete model is a $7 \times 11 \times 11 \times 11$ discrete Markov model of the dynamics, whose performance is shown in green. The error bars represent one standard error.

the trajectory that has zero probability under the evaluation policy π^m with a non-zero probability segment from the training data.

- A reduced order model provides both an approximation of the true dynamics and value function, which we can use to account for world stochasticity by correcting each trajectory's accumulated reward.

To implement the trajectory replacement insight, we first find any trajectory segment in the training data that has zero probability under π^m, our evaluation policy. The zero probability segment is then replaced with a randomly chosen segment from the training data that has a non-zero probability under π^m. Our second technique is implemented using the method of control variates. A control variate is a random variable that is correlated with a trajectory's accumulated reward and is used to reduce the variance of $\hat{V}(\pi^m; m_{true})$. Our control variate is formed using both the dynamics model m and $V(\pi^m; m)$, the value of operating π^m under m. We can additionally benefit from the property that the closer m is to m_{true} the greater the variance reduction. Once we obtain a on low variance estimate of $V(\pi^m; m_{true})$, we turn to choosing a model $m \in \hat{\mathcal{M}}$ that maximizes $\hat{V}(\pi^m; m_{true})$. To find this m we use a form of model likelihood gradient ascent.

One example problem is the hydrodynamic cart-pole problem, a fluid dynamics version of the cart-pole benchmark problem. We collected a data set of approximately 45 minutes of data from the hydrodynamic cart-pole system from a variety of poor controllers that was segmented into episodes with a maximum length of 10 seconds. This resulted in 316 episodes of batch training data.

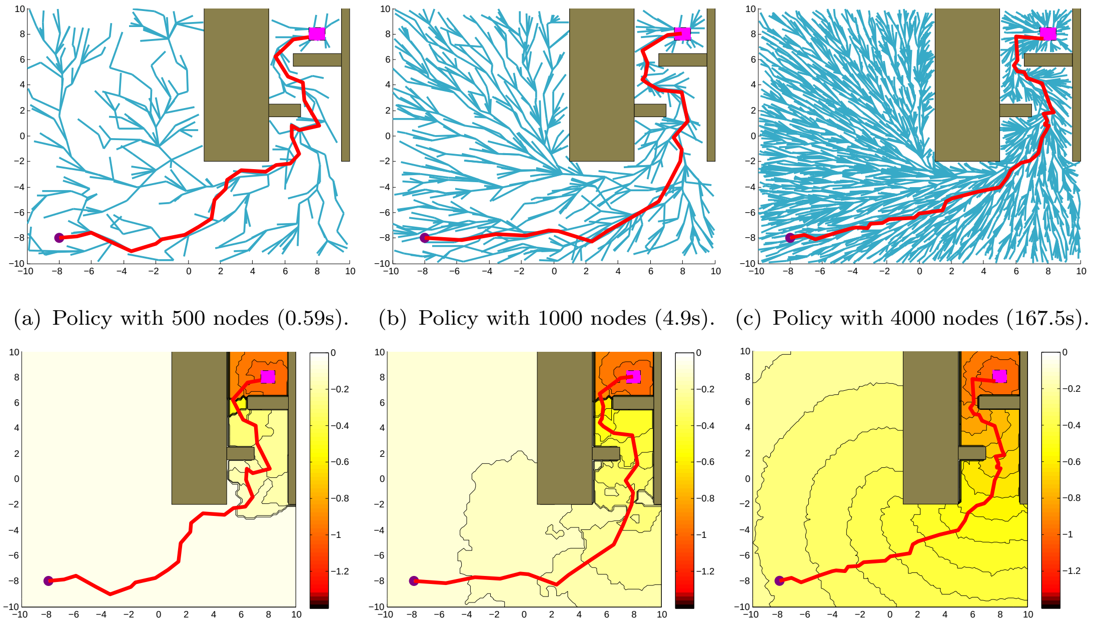
Figure 7 shows the results of this experiment plotted for the three different approaches. In red and blue are the returns of the approximate model class versus number of episodes with model selection performed by maximum likelihood and MRMS, respectively. Shown in green is the performance of the large Markov model composed of the same $7 \times 11 \times 11 \times 11$ grid as the value function and policy instead of the approximate model class.

For comparison purposes, the pole takes roughly 1-2 seconds to fall over, when given no control actions. The plot shows that for 45 minutes of data neither the maximum likelihood approximate model nor the large Markov model were able to achieve any statistically significant improvement in performance. Our approach, on the other hand, continually learned as it saw more data and eventually achieved the performance of stabilizing for all 10 of its trials after 316 episodes. Note that the error bars are drawn on the plot for MRMS's performance after 316 episodes of training.

In contrast to reduced-order models that have a finite size, our second approach is to use Bayesian nonparametric models for MBRL. Our previous work [28–30] showed that nonparametric approaches are well-suited for data-limited, poorly understood environments because they let the amount of training data determine the sophistication of the model and the Bayesian aspect helps the model to generalize to unseen data and also perform inference on noisy data.

5 Learning & Game-theoretic Approaches to Dynamic Vehicle Routing

A recurrent theme in all of the existing literature on dynamic vehicle routing (e.g., [31]) is that demands are either customers that need to be picked up, raw material or merchandise to be delivered, failures that must be serviced by a mobile repair person, or sites of suspicious activity that must be inspected. Thus far, the possibility of having an adversarial agent with limited capacity carry and place targets in the space from a central depot has not been considered. We model this problem and its inherent pure conflict of interests as a zero-sum game with two opponents: a vehicle moving at speed v that seeks to devise routing policies that minimize the average waiting time of a target, from the moment it is placed in the space until its location is visited; and an adversarial agent with unit capacity and speed, which aims at maximizing this time strategically choosing the process according to which he will place targets in the space. The fact that the agent has finite capacity, and therefore needs to return to the depot between successive rounds of target placements, induces a dependence between the temporal rate and the spatial distribution of targets. We analyze the game under light and heavy load regimes, and characterize the equilibria. The latter is the most interesting, for which we show that a TSP-based routing policy and a power-law spatial distribution are optimal. The latter emerges as the unique optimum of the problem of maximizing a nowhere differentiable convex integral functional over the space of probability distributions, which is solved using Fenchel duality [32, 33] and a logarithmic transformation to decouple the spatio-temporal dependence. Finally, we prove that the game has a value under any regime, and provide an expression for it as a function of v. Our results match and theoretically justify the so-called Rossmo's formula [34], an empirically-determined law which is at the basis of geographic profiling, a criminal investigative methodology that analyzes the locations of a connected series of crimes to determine the most probable area of offender residence.



(a) Policy with 500 nodes (0.59s). (b) Policy with 1000 nodes (4.9s). (c) Policy with 4000 nodes (167.5s).

(d) Contour of $\widetilde{J}_{500}$ (e) Contour of $\widetilde{J}_{1000}$ (f) Contour of $\widetilde{J}_{4000}$

Fig. 8: System with stochastic single integrator dynamics in cluttered environment. With appropriate cost structure assigned to the goal and obstacle regions, system reaches goal in upper right corner and avoids obstacles. The standard deviation of noise in x and y directions is 0.26. Anytime control policies in Fig. 8(a)-8(c) indicate that iMDP quickly explores the state space and refines control policies over time. Corresponding contours of approximated cost-to-go are shown in Fig. 8(d)-8(f).

6 Incremental Sampling-based Algorithms for Planning and Learning Under Uncertainty

In this effort, we consider a class of continuous-time, continuous-space stochastic optimal control problems. Building on recent advances in sampling-based algorithms for deterministic path planning [35], we propose a novel algorithm called the incremental Markov Decision Process (iMDP) to incrementally compute control policies that approximate the optimal policy with arbitrary well accuracy in terms of the expected cost. The main idea behind the algorithm is to generate a sequence of finite discretizations of the original problem through random sampling of the state space. At each iteration, the discretized problem is a Markov Decision Process that serves as an incrementally refined model of the original problem. We show that (i) the sequence of value functions induced by the optimal control policies for each of the discretized problems converge to the value function of the original stochastic optimal control problem, and (ii) the value function of the MDP at each iteration can be computed efficiently in an incremental manner. The proposed algorithm provides an anytime approach to the computation of optimal control policies, both for simulation-based or (reinforcement) learning-based approaches.

7 Strategic Dynamic Vehicle Routing with Spatio-Temporal Dependent Demands

In the recent past, considerable efforts have been devoted to the study of dynamic vehicle routing problems, where the objective is to cooperatively assign and schedule demands among a team of vehicles for service requests that are realized in a dynamic fashion over a region of interest [36, 37]. Throughout the existing literature, demands are assumed to be generated over time by an exogenous process that is unaffected by the routing policies, and in particular is non-adversarial [38]. However, there are many scenarios (e.g. surveillance missions) where there is an inherent conflict of interest between the process generating demands and the system planner designing routing policies. Moreover, even in non-adversarial scenarios the system planner may not have perfect information about the underlying process generating demands and a study of strategic dynamic vehicle routing can add insight into policies that are robust with respect to such uncertainty. To the best of our knowledge, settings with these characteristics have not yet been studied.

In this work [39] we consider the following problem: a system planner seeks to design dynamic routing policies for a team of vehicles that minimize the average waiting time of a typical demand, defined as the time difference between the moment the demand is placed in the region until its location is visited by a vehicle; while an adversarial agent with unitary capacity operating from a depot, aims at the opposite, strategically choosing the spatio-temporal stochastic process of demands. A novel feature of this setup is that, since demand generation is tied to the motion of the agent, there is a dependence between the spatial and temporal aspect of the demand generation process: the point process is thus completely specified by the spatial distribution. This is in stark contrast with the conventional setup for dynamic vehicle routing problems, where the spatial and temporal components of the demand generation process are typically assumed to be independent. We model the problem as a zero-sum game with two players: the system planner and the adversarial agent, with the average system time being the utility function. In the limiting regime where vehicles travel arbitrarily slower than the adversarial agent, we show that the game has a finite value and we characterize an equilibrium (or saddle point) of the game. This saddle point is shown to consist of a routing policy performing successive traveling salesperson (TSP) tours through outstanding demands and a unique power-law spatial density centered at the depot location. The saddle point routing policy is the one proposed in [37], where it is shown to be optimal for the setup where the demands are generated by an arbitrary spatio-temporal renewal process with a very high arrival rate. In order to rigorously determine the saddle point spatial distribution for the adversary we rely on Fenchel (conjugate) duality [40] and results from [33, 41] concerning the maximization of concave integral functionals subject to linear equality constraints. Since lower bounds for the average system time for dynamic vehicle routing under heavy load often take the form of concave integral functionals, the convex analytic approach applied in this work could be used more generally to formally analyze the performance of policies under worst case scenarios. Finally, incorporating the estimation of the spatial density into the problem could provide a natural framework for the formal study of geographic profiling [42], where the objective is to determine the most probable area of a predator's hideout ("anchor point") based on observed attack locations.

8 Learning in Persistent Search: A Constrained Game Approach

We consider a class of dynamic vehicle routing problems, known as persistent search [9], in which a vehicle with limited sensor range aims to detect targets that arrive dynamically over time. A common assumption made in such settings [10–12] is that the distribution of arrivals is known a priori to the searcher. Both the analysis and tools that have been developed previously have heavily relied on the knowledge of the distribution. To bridge the gap between optimistic assumptions and what can be done in practice, it is often suggested that the distribution be replaced with its empirical value [12], and hope that the results extend. In our recent work [8], we revisit these assumptions from a critical point of view, and argue that in practice, even if the targets are i.i.d samples from the same distribution over a compact support, one cannot learn the distribution at a rate that is fast enough in the sense that the gap between nominal performance index (assuming the knowledge of distribution) and that obtained by empirical mean is infinite. This is a direct consequence of the results in [43], which asserts that a non-parametric distribution cannot be learned at a rate faster than $\Omega(1/n)$ where n is the number of samples. Since the number of samples grows linearly with time, the accumulated error of estimation diverges. This implies that the performance index diverges almost surely from its nominal value since it depends somehow linearly on the quality of estimation.

The above results imply that it is not possible to provide any competitive analysis with respect to a fully informed algorithm that knows the distribution. Nevertheless, one can provide performance guarantees by modeling the problem in the language of constrained games. To do so, we assume that the searcher is playing a game with nature in which the samples are distributed in an adversarial manner that is consistent with data. In other words, we find the worst-case model of nature that can explain the past observations and find a search policy that is optimal against such adversarial model. The resulting optimization objective is non-convex, but is nevertheless separable [44] and can be approximated arbitrarily well with SOS methods [13]. The resulting solution can, in the presence of a smooth distribution that generates the data, guarantee a minimum level of performance.

Having established an effective method for solving the described constrained game, we move to an online setting where the vehicle is allowed to update its policy over time as data becomes available. Naturally, a policy in such settings serves two goals: information acquisition and target detection. Not surprisingly, whether a composition of optimal planner and empirical distribution is optimal depends on how close the two goals are. For example, if the goal is to minimize the average detection time the two goals are aligned and thus, the naive composition of optimal planner and estimator is provably competitive against what any other algorithm can do. However, if the goal was to minimize the sum of waiting times of arrivals until detection, then such compositions would not be close to optimal. One advantage of the SOS approach is that it allows us to compute near optimal policies even in cases where information acquisition and service goals tend to compete.

So far our approach has been model based learning, in the sense that we assume the existence of an oracle that generates i.i.d samples from a smooth function and reveals them to the searcher over time. We plan to relax this assumption and move to a setting where we can learn the policy directly from data. For that, however, we need to fix a hypothesis class that is rich enough to include the optimal policy. This is an indispensable ingredient of any learning mechanism. If one conjectures that the optimal policy is smooth (which is a weaker assumption than assuming underlying distribution is smooth and much weaker than assuming the underlying distribution is known), then it should be possible to approximate the optimal policy arbitrarily well with SOS polynomials over a compact region. We plan to test this conjecture for a class of DVR problems including persistent search.

Acknowledgment/Disclaimer: Work sponsored by the AFOSR, USAF, under grant # FA9550-09-1-0522. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFOSR or the U.S. Government.

Bibliography

[1] A. Geramifard, F. Doshi, J. Redding, N. Roy, and J. How, "Online discovery of feature dependencies," in *International Conference on Machine Learning (ICML)* (L. Getoor and T. Scheffer, eds.), (New York, NY, USA), pp. 881–888, ACM, June 2011.

[2] A. Geramifard, *Practical Reinforcement Learning Using Representation Learning and Safe Exploration for Large Scale Markov Decision Processes*. PhD thesis, Massachusetts Institute of Technology, November 2011.

[3] A. Geramifard, J. Redding, N. Roy, and J. P. How, "UAV Cooperative Control with Stochastic Risk Models," in *American Control Conference (ACC)*, pp. 3393 – 3398, June 2011.

[4] J. Redding, A. Geramifard, H.-L. Choi, and J. P. How, "Actor-Critic Policy Learning in Cooperative Planning," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2010. (AIAA-2010-7586).

[5] J. Redding, A. Geramifard, A. Undurti, H. Choi, and J. How, "An intelligent cooperative control architecture," in *American Control Conference (ACC)*, (Baltimore, MD), pp. 57–62, July 2010.

[6] A. Geramifard, J. Redding, J. Joseph, and J. P. How, "Model Estimation Within Planning and Learning," in *Workshop on Planning and Acting with Uncertain Models, ICML, Bellevue, WA, USA*, June 2011.

[7] A. Geramifard, J. Redding, J. Joseph, N. Roy, and J. P. How, "Model estimation within planning and learning," in *American Control Conference (ACC)*, June 2012 (too appear).

[8] H. Roozbehani and E. Frazzoli, "Persistent search with unknown distribution," in *2012 IEEE Conference on Decision and Control (in preparation)*, dec. 2012.

[9] V. A. Huynh, J. Enright, and E. Frazzoli, "Persistent patrol with limited-range on-board sensors," in *Decision and Control (CDC), 2010 49th IEEE Conference*, pp. 7661 –7668, dec. 2010.

[10] D. P. Bertsekas and D. A. Castanon, "Parallel synchronous and asynchronous implementations of the auction algorithm," *Parallel Computing*, vol. 17, pp. 707–732, 1991.

[11] J. Enright, *Efficient Routing of Multi-Vehicle Systems: limited sensing and nonholonomic motion constraint*. PhD thesis, UCLA, 2008.

[12] M. Pavone, *Dynamic Vehicle Routing for Robotic Networks*. PhD thesis, MIT, 2010.

[13] P. Parrilo, "Polynomial games and sum of squares optimization," in *Decision and Control, 2006 45th IEEE Conference*, pp. 2855 –2860, dec. 2006.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[15] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems 8*, pp. 1038–1044, The MIT Press, 1996.

[16] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.

[17] S. Mahadevan, "Representation policy iteration," *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

[18] R. Parr, C. Painter-Wakefield, L. Li, and M. Littman, "Analyzing feature generation for value-function approximation," in *International Conference on Machine Learning (ICML)*, (New York, NY, USA), pp. 737–744, ACM, 2007.

[19] S. Sanner, "Online feature discovery in relational reinforcement learning," in *Proceedings of the Open Problems in Statistical Relational Learning Workshop*, 2006.

[20] S. Whiteson, M. E. Taylor, and P. Stone, "Adaptive tile coding for value function approximation," Tech. Rep. AI-TR-07-339, University of Texas at Austin, 2007.

[21] B. Ratitch and D. Precup, "Sparse distributed memories for on-line value-based reinforcement learning," in *European Conference on Machine Learning (ECML)*, pp. 347–358, 2004.

[22] N. D. Goodman, J. B. Tenenbaum, T. L. Griffiths, and J. Feldman, "Compositionality in rational analysis: Grammar-based induction for concept learning," in *The probabilistic mind: Prospects for Bayesian cognitive science* (M. Oaksford and N. Chater, eds.), 2008.

[23] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, pp. 912–926, August 2009.

[24] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Incremental natural actor-critic algorithms.," in *Advances in Neural Information Processing Systems (NIPS)* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 105–112, MIT Press, 2007.

[25] G. A. Rummery and M. Niranjan, "Online Q-learning using connectionist systems (tech. rep. no. cued/f-infeng/tr 166)," *Cambridge University Engineering Department*, 1994.

[26] R. I. Brafman and M. Tennenholtz, "R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 213–231, 2001.

[27] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst, "Model-free monte carlo-like policy evaluation," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 217–224, 2010.

[28] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A Bayesian nonparametric approach to modeling motion patterns," *Autonomous Robots*, no. 2, pp. 123–134, 2011 (Accepted, awaiting publication). Search and Pursuit/Evasion with Mobile Robots.

[29] J. Joseph, F. Doshi-Velez, and N. Roy, "A Bayesian nonparametric approach to modeling mobility patterns," in *AAAI'10*, AAAI Press, 2010.

[30] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A Bayesian nonparametric approach to modeling motion patterns," tech. rep., 2011.

[31] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. Smith, "Dynamic vehicle routing for robotic systems," *Proceeedings of the IEEE*, 2011. To appear.

[32] R. T. Rockafellar, "Integrals which are convex functionals," *Pacific Journals of Mathematics*, vol. 24, no. 3, 1968.

[33] J. M. Borwein and A. S. Lewis, "Duality relationships for entropy-like minimization problems," *SIAM J. Control and Optimization*, vol. 29, pp. 325–338, March 1991.

[34] K. D. Rossmo, *Geographic profiling: target patterns of serial murderers.* PhD thesis, Simon Fraser University, 1995.

[35] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. Journal of Robotics Research*, 2011. To appear.

[36] D. J. Bertsimas and G. J. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, pp. 601–615, 1991.

[37] D. J. Bertsimas and G. J. van Ryzin, "Stochastic and dynamic vehicle routing with general interarrival and service time distributions," *Advances in Applied Probability*, vol. 25, pp. 947–978, 1993.

[38] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. Smith, "Dynamic vehicle routing for robotic systems," *Proceeedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.

[39] K. S. D. Feijer and E. Frazzoli, "Strategic dynamic vehicle routing with spatio-temporal dependent demands," in *Proc. American Control Conference*, 2012. To Appear.

[40] R. T. Rockafellar, *Convex Analysis.* Princeton University Press, 1970.

[41] J. M. Borwein and A. S. Lewis, "Partially finite convex programming," *Mathematical Programming*, vol. 57, pp. 15–83, 1992.

[42] D. K. Rossmo, *Geographic Profiling.* CRC Press, 1999.

[43] D. W. Boyd and J. M. Steele, "Lower bounds for nonparametric density estimation rates," *Ann. Statist.*, vol. 6, no. 4, pp. 932–934, 1978.

[44] N. D. Stein, A. Ozdaglar, and P. A. Parrilo, "Separable and low-rank continuous games," *Internat. J. Game Theory*, vol. 37, no. 4, pp. 475–504, 2008.

Personnel: Professors– How, Frazzoli, and Roy; Grad– Alborz Geramifard, Hajir Roozbehani, Diego Feijer, Vu Anh Huynh (Partial), Joshua Joseph and Adam Bry (partial)

Recent publications: See [1–8, 28–30]

Honors/Awards:

- Professor How was awarded the Richard Cockburn Maclaurin Professor of Aeronautics and Astronautics in July 2009.
- "Best Applications paper published in Automatica over the past three years" in August 2011.

AFRL Points of Contact: Dr. Fariba Fahroo, AFOSR/RSL and Dr. Banda (AFRL/RBCA).

Transitions: Air force SBIR-STTR document number AF121-002 with title "Intelligent Controller Development for Cooperative UAV Missions", cited the iCCA work [5] as one of the main references.